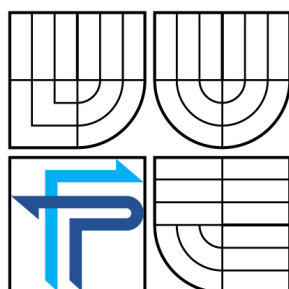


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

APLIKACE PRO ZPRACOVÁNÍ A VIZUALIZACI ČASOVÝCH ŘAD

AN APPLICATION FOR PROCESSING AND VISUALIZATION OF TIME SERIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ DIERZÉ

VEDOUcí PRÁCE
SUPERVISOR

Ing. KAREL DOUBRAVSKÝ, Ph.D.

BRNO 2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Dierzé Lukáš

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Aplikace pro zpracování a vizualizaci časových řad

v anglickém jazyce:

An Application for Processing and Visualization of Time Series

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy


Seznam odborné literatury:


- ANDĚL, J. Základy matematické statistiky. 2. vyd. Praha : Matfyzpress, 2007. ISBN 978-80-7378-001-2.
- CIPRA, T. Analýza časových řad s aplikacemi v ekonomii. 1. vyd. Praha : SNTL/ALFA, 1986. ISBN 99-00-00157-X.
- HINDLS, R. a HRONOVÁ, S. a SEGER, J. Statistika pro ekonomy. 1.vyd. Praha : Professional Publishing, 2002. ISBN 80-86419-26-6.
- KROPÁČ, J. Statistika B. 1. vyd. Brno : VUTFP, 2006. ISBN 80-214-3295-0.
- RYAN, T. P. Modern Regression Methods. New York : John Wiley&Sons, Inc., 1997. ISBN 0-471-52912-5.
- ZVÁRA, K. Regresní analýza. 1. vyd. Praha : Academia, 1989. ISBN 80-200-0125-5.

Vedoucí bakalářské práce: Ing. Karel Doubravský, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/10.




Ing. Jiří Kříž, Ph.D.
Ředitel ústavu


doc. RNDr. Anna Putnová, Ph.D., MBA
Děkanka

V Brně, dne 7. 2. 2010

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a realizací softwarové aplikace určené pro analýzu a vizualizaci ekonomických časových řad v podnikové sféře.

KLÍČOVÁ SLOVA

Časové řady, programování, regresní analýza, software, statistika

ABSTRACT

This bachelor's thesis deals with designing and creation of software application oriented for analysis and visualization of economic time series in company environment.

KEYWORDS

Time series, programming, regression analysis, statistics, software

DIERZÉ, L. *Aplikace pro zpracování a vizualizaci časových řad*. Brno : Vysoké učení technické v Brně, Fakulta podnikatelská. 2010. 52 s. Vedoucí bakalářské práce byl Ing. Karel Doubravský, PhD.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne

.....

(podpis autora)

Poděkování

Tímto bych rád poděkoval panu Ing. Karlu Doubravskému, Ph.D. za pomoc, věcné připomínky, rady a čas, který mi věnoval při zpracování této bakalářské práce.

OBSAH

ÚVOD	10
VYMEZENÍ PROBLÉMU A CÍL PRÁCE	11
1 TEORETICKÁ ČÁST	12
1.1 Regresní analýza	12
1.1.1 Úvod	12
1.1.2 Lineární regresní modely	13
1.1.3 Nelineární regresní modely	15
1.1.4 Metody klouzavých průměrů	17
1.2 Časové řady	19
1.2.1 Úvod	19
1.2.2 Grafická interpretace	20
1.2.3 Charakteristiky	21
1.2.4 Dekompozice časových řad	22
1.2.5 Popis trendu časové řady	24
1.2.6 Predikce	25
1.3 Softwarová aplikace	25
1.3.1 Základní předpoklady	25
1.3.2 Programovací jazyk	26
1.3.3 Nástroje	27
1.3.4 Použité technologie	28
2 IMPLEMENTACE APLIKACE	31
2.1 Vrstvy aplikace	31
2.1.1 Aplikační vrstva	31
2.1.2 Datová vrstva	31
2.1.3 Prezentační vrstva	32
2.2 Jmenné prostory a třídy	32
2.2.1 Thesis	33

2.2.2	Thesis.Core	33
2.2.3	Thesis.Core.Functions	36
2.2.4	Thesis.GUI	38
3	TESTOVÁNÍ APLIKACE	42
3.1	Testovací data	42
3.2	Vytvoření časové řady	42
3.3	Zpracování a vizualizace dat	43
3.3.1	Výpočet charakteristik a vyrovnání dat	43
3.3.2	Vykreslení grafu	44
3.4	Prognóza	44
3.5	Časová náročnost výpočtu	46
4	NÁVRHY NA ROZŠÍŘENÍ	47
	ZÁVĚR	48
	LITERATURA	49
	SEZNAM OBRÁZKŮ	50
	SEZNAM TABULEK	51
	SEZNAM POUŽITÝCH ZKRATEK	52
	SEZNAM PŘÍLOH	53

ÚVOD

Cílem mé bakalářské práce bylo navrhnout a vytvořit softwarovou aplikaci pro analytické zpracování a vizualizaci časových řad užitím statistických metod a tím zpřístupnit rozsáhlé možnosti těchto metod při vyhodnocování ekonomických dat v podniku.

První část této práce se věnuje teorii z oblasti statistiky, zejména pak regresní analýzy a analýzy časových řad, neboť právě tyto oblasti jsou pro využití v podnikové sféře nejpodstatnější.

Ve druhé části práce je popsána implementace softwarové aplikace s popisem vytvořených a použitých tříd a jejich metod. Poté následuje kapitola zabývající se otestováním aplikace.

VYMEZENÍ PROBLÉMU A CÍL PRÁCE

Vymezení problému

V dnešní době snad v každém podniku dochází k tvorbě plánů většinou na základě zkušeností či ekonomické analýzy finančních dat, ať už okamžikových nebo historických. U těchto ekonomických ukazatelů, které jsou povětšinou všem manažerům dobře známy, se velice často pracuje s daty, která jsou na sobě různě závislá a jejich pouhé dosazení do daných vzorců není ideální cestou k relevantní analýze.

V ekonomickém hodnocení společnosti se tak velice často pracuje s proměnnými veličinami, kdy mezi měřenými ukazateli existuje jistá, námi hledaná, závislost. Proto je vhodné u těchto analýz využívat i specifických statistických metod k eliminaci možných nežádoucích chyb v hodnocení situace podniku.

Cíl práce

Cílem mé bakalářské práce je navrhnout a vytvořit aplikaci pro analýzu časových řad a tím zpřístupnit využití potřebných statistických metod pro neodbornou veřejnost pomocí softwarové aplikace, např. pro manažery bez hlubší znalosti matematiky a statistiky.

1 TEORETICKÁ ČÁST

1.1 Regresní analýza

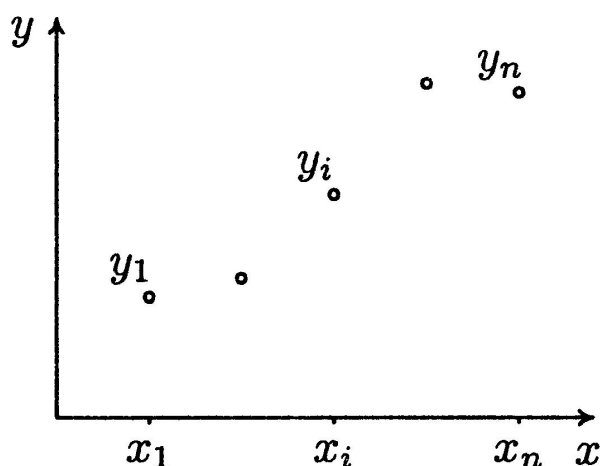
1.1.1 Úvod

Snad ve všech ekonomických funkcích a vztazích se setkáváme s různými proměnnými veličinami, mezi kterými vždy existuje blíže neurčená závislost. Obvykle se jedná o nezávisle proměnnou x a závisle proměnnou y , mezi kterými panuje nám zatím nepopsaná závislost. Tuto závislost můžeme obecně zapsat tímto předpisem :

$$y = \varphi(x). \quad (1.1)$$

Účelem regresní analýzy je tedy co možná nejlépe popsat vzájemnou závislost mezi těmito veličinami, a to pomocí vhodné regresní funkce $\varphi(x)$. [5]

Měření těchto neznámých veličin probíhá tak, že vždy zkoumáme hodnoty závisle proměnné y při různých hodnotách nezávisle proměnné x . Provedením dvou a více měření pak dostaneme n dvojic (x_i, y_i) , kde x_i značí nastavenou hodnotu v i -tém měření a y_i k ní naměřenou hodnotu. Tato situace je znázorněna na následujícím obrázku.



Obrázek 1.1: Znázornění naměřených n dvojic, Zdroj: [5], str. 79

Každé měření je však ovlivněno tzv. „šumem“ vznikajícím působením náhodných okolních vlivů a předem neidentifikovatelných činitelů, díky čemuž dostaneme při každém měření pro stejnou hodnotu proměnné x jinou, většinou velice blízkou hodnotu proměnné y . Závisle proměnnou y tedy můžeme klasifikovat jako náhodnou veličinu a označíme ji Y . Z předchozího vyplývá, že závislost mezi proměnnými x a y je ovlivněna i „šumem“, který je však, pokud zanedbáme systematické chyby měření, svým způsobem také náhodná veličina, kterou značíme e . Předpokladem správného měření (tj. bez chyb) je skutečnost, že střední hodnota této náhodné veličiny je rovna nule, tedy $E(e)=0$ a výchyly způsobené šumem jsou od skutečných hodnot rozloženy na kladné i záporné straně. [5]

Pro popsání závislosti náhodné veličiny Y na proměnné x zavedeme tzv. *podmíněnou střední hodnotu náhodné veličiny Y pro hodnotu x* , označenou $E(Y|x)$ a položíme ji rovnu vhodné *regresní funkci* $\eta(x)$ s neznámými parametry β a tuto zapíšeme $\eta(x; \beta_1, \beta_2, \dots, \beta_p)$, kde $p \geq 1$. Závislost mezi střední hodnotou „šumu“ a regresní funkcí lze vyjádřit předpisem :

$$E(Y | x) = \eta(x; \beta_1, \beta_2, \dots, \beta_p). \quad (1.2)$$

1.1.2 Lineární regresní modely

Mezi tzv. „lineární regresní modely“ patří ty regresní funkce, jejichž parametry jsou lineární nebo je lze pomocí vhodných matematických transformací linearizovat. Tyto modely jsou obecně vyjádřeny následujícím předpisem :

$$\eta(x) = x_1\beta_1 + x_2\beta_2 + \dots + x_n\beta_n. \quad (1.3)$$

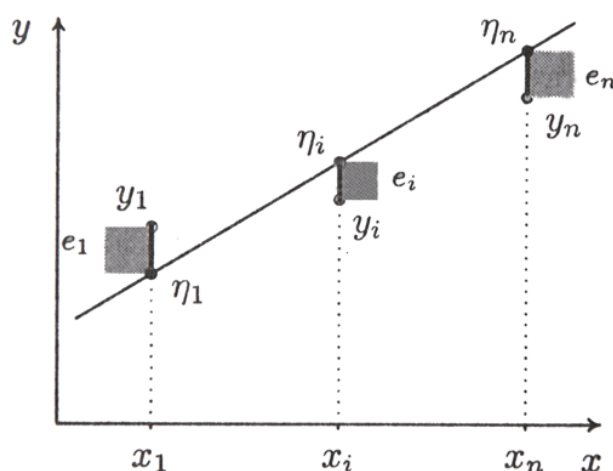
Regresní přímka

Jedním z nejčastějších případů lineární regresní funkce, se kterou se většinou setkáváme, je tzv. *regresní přímka*, která je vyjádřena tímto předpisem :

$$\eta(x) = \beta_1 + \beta_2 x \quad (1.4)$$

Důkazem lineárnosti regresní přímky je splnění obecného předpisu lineárních regresních funkcí (1.3) a platí, že náhodná veličina Y je rovna součtu příslušné regresní funkce $\eta(x)$ a „šumu“ e . Pro stanovení odhadu regresní přímky pro naměřené dvojice dat (x_i, y_i) je proto důležité správně stanovit koeficienty b_1 a b_2 . K tomu poslouží metoda nejmenších čtverců ilustrovaná na obr. (1.2), jejímž využitím stanovíme následující předpis funkce $S(b_1, b_2)$

$$S(b_1, b_2) = \sum_{i=1}^n (y_i - b_1 - b_2 x_i)^2. \quad (1.5)$$



Obrázek 1.2: Metoda nejmenších čtverců, Zdroj: [5], str. 80

Vypočtením první parciální derivace funkce $S(b_1, b_2)$ z předpisu funkce (1.5) podle obou proměnných b_1 a b_2 získáme rovnice, jejichž další úpravou dostaneme následující *soustavu normálních rovnic* :

$$\begin{aligned} n \cdot b_1 + \sum_{i=1}^n x_i \cdot b_2 &= \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i \cdot b_1 + \sum_{i=1}^n x_i^2 \cdot b_2 &= \sum_{i=1}^n x_i y_i \end{aligned} \quad (1.6)$$

z níž koeficienty b_1 a b_2 můžeme vypočítat prakticky jakoukoli známou metodou pro počítání rovnic, avšak pro další využití těchto koeficientů v této práci je vypočteme podle vzorců (1.7).

$$b_2 = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sum_{i=1}^n x_i^2 - n \bar{x}^2}, \quad (1.7)$$

kde \bar{x}, \bar{y} jsou výběrové průměry, které vypočítáme pomocí předpisů

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (1.8)$$

Po stanovení těchto koeficientů můžeme určit odhad regresní přímky, označený $\hat{\eta}(x)$ a vyjádřit ho předpisem

$$\hat{\eta}(x) = b_1 + b_2 x. \quad (1.9)$$

1.1.3 Nelineární regresní modely

Mezi tzv. „nelineární regresní modely“ patří ty regresní funkce, které jsou vyjádřeny nelineárně v parametrech a ani je zpravidla nelze pomocí matematických transformací linearizovat. Příkladem nelineárního regresního modelu může být např. funkční předpis (1.10). Existují však určité nelineární regresní modely, které lze užitím vhodných matematických transformací převést na lineární, potom hovoříme o tzv. „linearizaci“.

$$\eta(x) = \beta_1 + \beta_e e^{\beta_3 x} \quad (1.10)$$

Linearizovatelné funkce

„Říkáme, že nelineární regresní funkce $\eta(x, \beta)$ ji *linearizovatelná*, jestliže vhodnou transformací dostaneme funkci, která na svých regresních koeficientech závisí lineárně. Pro určení regresních koeficientů a dalších charakteristik této linearizované funkce použijeme buď regresní přímku, nebo klasický lineární model. Zpětnou transformací pak ze získaných výsledků dostaneme odhady koeficientů a dalších charakteristik pro nelineární model“ ([5], str. 104)

Příkladem takové funkce může být tzv. „exponenciální trend“, jehož nelineárnost lze převést na funkci lineární v parametrech pomocí logaritmování. Tuto funkci zapisujeme předpisem :

$$\eta(x) = \beta_1 \beta_2^x, \text{ kde } x = 1, 2, \dots, n. \quad (1.11)$$

Následujícím postupem získáme z funkce (1.11) nový tvar, který je velice podobný předpisu regresní přímky (1.4).

$$\begin{aligned} \eta(x) &= \beta_1 \beta_2^x & / \ln \\ \ln \eta(x) &= \ln \beta_1 + x \ln \beta_2 \end{aligned} \quad (1.12)$$

Nyní provedeme jednoduchou substituci, kdy za $\ln \eta(x)$ dosadíme z , za $\ln \beta_1$ dosadíme c_1 a za $\ln \beta_2$ dosadíme c_2 :

$$z = c_1 + x c_2 \quad (1.13)$$

S takto upraveným exponenciálním trendem už můžeme pracovat jako s regresní přímkou, přičemž po získání koeficientů tyto zpětně dosadíme do původní rovnice (1.11). [4]

Nelinearizovatelné funkce

Jak již bylo řečeno v úvodu této podkapitoly, tyto funkce jsou vyjádřeny nelineárně v parametrech a ani je na lineární v parametrech nelze matematicky transformovat. Následující příklady jsou případy speciálních nelinearizovatelných funkcí užívaných zejména při analýze časových řad. [5]

• Modifikovaný exponenciální trend

$$\eta(x) = \beta_1 + \beta_2 \beta_3^x \quad (1.14)$$

Modifikovaný exponenciální trend se používá, když je regresní funkce shora nebo zdola ohraničená.

- **Logistický trend**

$$\eta(x) = \frac{1}{\beta_1 + \beta_2 \beta_3^x} \quad (1.15)$$

Logistický trend je shora i zdola ohraničen a průběh jeho křivky se mění v inflexním bodě a patří mezi tzv. *S-křivky* symetrické kolem inflexního bodu.

- **Gompertzova křivka**

$$\eta(x) = e^{\beta_1 + \beta_2 \beta_3^x} \quad (1.16)$$

Gompertzova křivka je shora i zdola ohraničena a její průběh se mění v inflexním bodě a patří mezi tzv. *S-křivky* nesymetrické kolem inflexního bodu, protože většina jejích hodnot leží až za tímto bodem.

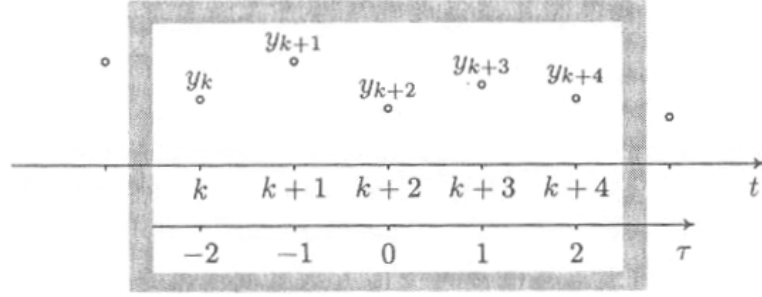
S-křivky jsou speciálními případy křivek používaných zejména v ekonomii k modelování ekonomických situací. Jejich podstata spočívá v tom, že s jejich pomocí lze zobrazit na časové ose pět základních fází ekonomického cyklu. [5]

1.1.4 Metody klouzavých průměrů

„Metody klouzavých průměrů, jichž je několik, se používají pro popis trendu v časové řadě, který mění v čase svůj charakter a pro jehož popis nelze použít vhodnou matematickou funkci.“ ([5], str. 125)

V této práci se budu zabývat vyrovnaním hodnot pomocí *vážených klouzavých průměrů*, jejichž princip spočívá v proložení nejprve první pěti hodnot časové řady polynomem třetího stupně, kdy určíme vyrovnané první tři hodnoty z této pěti a poté se od prostřední hodnoty posuneme k další pěti hodnot, kde postupně určíme vyrovnané hodnoty prostředních hodnot. Na závěr vyrovnáme poslední dvě hodnoty z poslední pěti.

Předpokladem použití této metody je časová řada y_1, y_2, \dots, y_n , kde každou pěti hodnot můžeme znázornit tzv. *okénkem pro klouzavé průměry* (obr. 1.3), kde si



Obrázek 1.3: Okénko pro klouzavé průměry, Zdroj: [5], str. 126

pro přehlednost výpočtu definujeme časové úseky $k, k+1, k+2, k+3, k+4$ jako hodnoty $-2, -1, 0, 1, 2$ na pomocné ose τ . Poté tyto hodnoty proložíme následujícím regresním polynomem třetího stupně:

$$\eta(\tau, \mathbf{b}_k) = b_{k1} + b_{k2}\tau + b_{k3}\tau^2 + b_{k4}\tau^3. \quad (1.17)$$

kde \mathbf{b}_k je sloupcový vektor (vz. 1.19) jeho koeficientů, jejichž hodnoty vypočteme pomocí již výše zmiňované metody nejmenších čtverců, kdy zjistíme minimum této funkce :

$$S(b_{k1}, b_{k2}, b_{k3}, b_{k4}) = \sum_{\tau=-2}^2 (y_{k+2+\tau} - b_{k1} - b_{k2}\tau - b_{k3}\tau^2 - b_{k4}\tau^3). \quad (1.18)$$

Vypočtením parciálních derivací podle regresních koeficientů a položením získaných výrazů nule získáme soustavu čtyř rovnic o čtyřech neznámých, kterou lze vyjádřit v maticovém tvaru $\mathbf{A} \cdot \mathbf{b}_k = \mathbf{c}_k$, kde \mathbf{c}_k je sloupcový vektor vypočtený z hodnot časové řady a hodnot definované proměnné τ k -tého okénka. Jednotlivé matice tedy jsou :

$$\mathbf{A} = \begin{bmatrix} 5 & 0 & 10 & 0 \\ 0 & 10 & 0 & 34 \\ 10 & 0 & 34 & 0 \\ 0 & 34 & 0 & 130 \end{bmatrix}, \mathbf{b}_k = \begin{bmatrix} b_{k1} \\ b_{k2} \\ b_{k3} \\ b_{k4} \end{bmatrix}, \mathbf{c}_k = \begin{bmatrix} \sum y_{k+2+\tau} \\ \sum \tau \cdot y_{k+2+\tau} \\ \sum \tau^2 \cdot y_{k+2+\tau} \\ \sum \tau^3 \cdot y_{k+2+\tau} \end{bmatrix} \quad (1.19)$$

Koeficienty $b_{k1}, b_{k2}, b_{k3}, b_{k4}$ určíme z rovnice :

$$\mathbf{b}_k = \mathbf{A}^{-1} \mathbf{c}_k. \quad (1.20)$$

Úpravou a dosazením do tohoto vzorce odvodíme následující předpis pro vyrovnaní prostřední hodnoty y_{k+2} v k -tém okénku :

$$\hat{\eta}_{k+2} = \frac{1}{35}(-3 \cdot y_k + 12 \cdot y_{k+1} + 17 \cdot y_{k+2} + 12 \cdot y_{k+3} - 3 \cdot y_{k+4}) \quad (1.21)$$

Pro vyrovnaní prvních, resp. posledních dvou hodnot nejprve spočteme požadované vektory \mathbf{b}_k a \mathbf{c}_k , kde $k = 1$ pro první dvě hodnoty a $k = n - 4$ pro poslední dvě hodnoty a poté hodnoty dosadíme do rovnice (1.17), kde za proměnnou τ dosadíme hodnoty $-2, -1$ pro první okénko, resp. $1, 2$ pro poslední okénko, čili dostaneme následující rovnice:

$$\hat{\eta}_1 = \hat{\eta}(-2, \mathbf{b}_1); \quad \hat{\eta}_1 = \hat{\eta}(-1, \mathbf{b}_1) \quad (1.22)$$

$$\hat{\eta}_{n-1} = \hat{\eta}(1, \mathbf{b}_{n-4}); \quad \hat{\eta}_n = \hat{\eta}(2, \mathbf{b}_{n-4}) \quad (1.23)$$

1.2 Časové řady

1.2.1 Úvod

„Časovou řadou (někdy chronologickou řadou) rozumíme řadu hodnot určitého ukazatele, uspořádaných z hlediska přirozené časové posloupnosti. Přitom je nutné, aby věcná náplň ukazatele i jeho prostorové vymezení byly shodné v celém sledovaném úseku.“ ([5], str. 114)

Díky této posloupnosti dat srovnaných od minulosti do současnosti (budoucnosti) můžeme pomocí vhodné analýzy sledovat buď současný trend, nebo prognózovat budoucí vývoj. Při analýze dat je třeba pamatovat na to, že se časové řady rozdělují podle různých kritérií. Pro účely této práce postačí rozdělení z hlediska času.

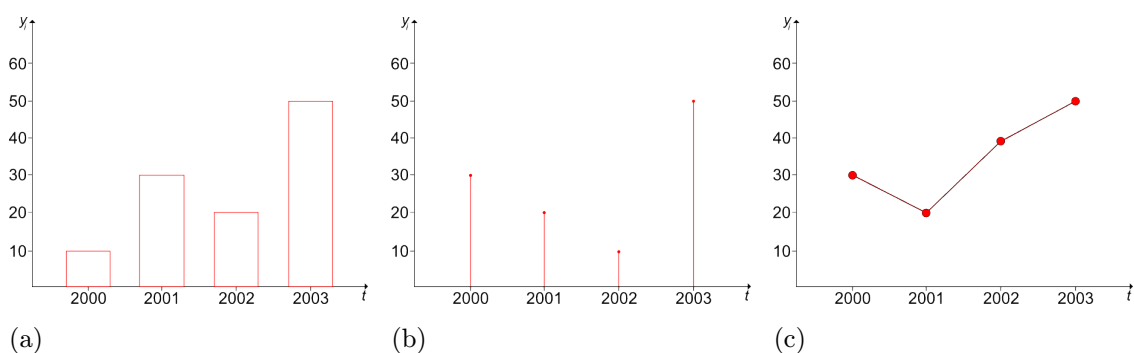
Jedná se o časové řady *intervalové*, tj. ty, které zaznamenávají nějaký jev, stav v určitém období (např. vývoj cen pohonných hmot v průběhu jednoho kalendářního roku) a na *okamžikové*, tj. zaznamenávající nastalý jev, či stav, v určitý okamžik (např. ceny paliv k 1.1.2010). [5]

Zásadním rozdílem mezi těmito typy je, že intervalové časové řady lze pro potřeby statistiky s úspěchem sčítat, kdežto sčítání okamžikových (tedy izolovaných dat) nemá logický význam. [5]

1.2.2 Grafická interpretace

Pro lepší interpretaci samotných dat a výsledků analýzy je přinejmenším vhodné zobrazit data graficky. Intervalové časové řady je možné zobrazit pomocí grafů

- **sloupcových** (obr. 1.4a),
- **hůlkových** (obr. 1.4b),
- **spojnicových** (obr. 1.4c).



Obrázek 1.4: Sloupcový, hůlkový a spojnicový graf, Zdroj: vlastní

Okamžikové časové řady lze tak vzhledem k jejich povaze zobrazit výhradně pomocí spojnicových grafů (obr. 1.4c).

1.2.3 Charakteristiky

Předpokládáme, že hodnoty v časových okamžicích, resp. intervalech t_i , kde $i=1,2,\dots,n$, označené y_i , jsou při výpočtech následujících charakteristik vždy kladné a že intervaly mezi sousedními časovými okamžiky nebo středy časových intervalů jsou stejně dlouhé.

Průměry

V případě *intervalových* časových řad se jedná o tzv. „aritmetický průměr“, který je dán vzorcem

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n, \quad (1.24)$$

přičemž v případě *okamžikových* časových řad se jedná o tzv. „chronologický průměr“. Pokud je ve vzdálenostech mezi jednotlivými časovými úseky zjevná posloupnost, nazýváme jej *neváženým chronologickým průměrem* a vypočteme jej pomocí vzorce :

$$\bar{y} = \frac{1}{n-1} \left[\frac{y_1}{2} + \sum_{i=2}^{n-1} y_i + \frac{y_n}{2} \right]. \quad (1.25)$$

První difference

Pro analýzu časových řad pak slouží další charakteristiky popisu vývoje časových řad. Mezi ty nejzákladnější patří tzv. „první difference“, které vyjadřují přírůstek oproti předchozí hodnotě, a počítáme je podle tohoto předpisu :

$${}_1d_i(y) = y_i - y_{i-1}, \quad kde \ i = 2, 3, \dots, n. \quad (1.26)$$

Pro lepší přehled je možné z prvních diferencí vypočítat i průměrný přírůstek za sledované období, tzv. „průměr prvních diferencí“, a to podle vzorce :

$$\overline{{}_1d_i(y)} = \frac{1}{n-1} \sum_{i=2}^n {}_1d_i(y) = \frac{y_n - y_1}{n-1}. \quad (1.27)$$

Průměrný koeficient růstu

Poslední významnou charakteristikou je tzv. „průměrný koeficient růstu“ udávající průměrnou rychlost růstu, či poklesu hodnot, přičemž z následujícího vzorce jasně vyplývá, že tato a předchozí charakteristika časové řady závisí pouze na její první a poslední hodnotě a jejich význam tedy přichází v úvahu pouze v případě jejího monotónního vývoje.

$$\overline{k(y)} = \sqrt[n-1]{\prod_{i=2}^n k_i(y)} = \sqrt[n-1]{\frac{y_n}{y_1}} \quad (1.28)$$

1.2.4 Dekompozice časových řad

Všechny hodnoty časové řady lze obecně rozložit (tj. dekomponovat) do čtyř základních složek, ve kterých pohled na data závisí na typu dekompozice. V případě *aditivní dekompozice* lze hodnoty y_i časové řady vyjádřit součtem

$$y_i = T_i + C_i + S_i + e_i, \quad (1.29)$$

kdežto při *multiplikativní dekompozici* jsou tyto hodnoty vyjádřeny součinem

$$y_i = T_i \cdot C_i \cdot S_i \cdot e_i. \quad (1.30)$$

V obou případech jednotlivým prvkům přiřkládáme následující význam :

- T_i = hodnota trendové složky,
- C_i = hodnota sezónní složky,
- S_i = hodnota cyklické složky,
- e_i = hodnota reziduální složky.

V prvním případě, tedy u *aditivní dekompozice*, jsou všechny tyto složky uvažovány ve svých skutečných absolutních hodnotách a jsou měřeny v jednotkách y_i . Avšak u *multiplikativní dekompozice* je ve své absolutní hodnotě uvažována pouze trendová složka a zbytek složek je uvažován v hodnotách relativních. [5]

Trendová složka

Trend obecně vyjadřuje jakýsi dlouhodobě trvající směr vývoje určitého ukazatele v čase. V případě, kdy se hodnoty sledovaného ukazatele v průběhu analyzovaného období příliš nemění a spíše kolísají okolo stejné hodnoty hovoříme o tzv. „časové řadě bez trendu“.

Sezónní složka

Tato složka určuje periodické změny v analyzované časové řadě, které se odehrávají během sledovaného období a každé toto stejně dlouho trvající období se opakuje.

„Sezónní změny jsou hlavně způsobeny takovými faktory, jako je střídání ročních období nebo lidské zvyky, spočívající v ekonomické aktivitě, např. změny v průměrných měsíčních teplotách nebo změny v objemu sezónního prodeje obchodního domu během roku.“ ([5], str. 123)

Cyklická složka

Tato složka je nejspornější částí celé dekompozice, neboť vliv cyklické složky na hodnoty časové řady lze těžko určit a může být zapříčiněn mnoha vnějšími, neurčitelnými vlivy. Z tohoto důvodu je také velice obtížné tuto složku v dekompozici eliminovat, neboť její věcný a matematický charakter v dané časové řadě je těžké určit.

Reziduální složka

V literatuře někdy označovaná jako tzv. „náhodná složka“ zůstane po odečtení všech „určitých“ složek a je tedy tvořena pouze hodnotami, které mají blíže neurčitelný systematický charakter. Z toho vyplývá, že se může jednat buď o nerozpoznatelné jevy nebo o chyby měření vzniklé ať už při samotném sběru nebo při následném zpracování dat.

1.2.5 Popis trendu časové řady

Nejčastějším způsobem v určování trendu analyzované časové řady je v předchozí kapitole zmiňovaná regresní analýza, díky které můžeme naměřená data časové řady nejen vyrovnat, ale velice často i prognózovat jejich budoucí vývoj.

Při popisu trendu je tak velice důležité zvolit vhodnou regresní funkci, přičemž „vhodnost“ posuzujeme buď z grafického zobrazení časové řady, nebo, v našem případě, na základě tzv. „indexu determinace“.

„Jedním z úkolů regresní analýzy je posouzení, zda zvolená regresní funkce je pro vyrovnání zadaných dat vhodná. Řešení této úlohy spočívá jednak ve zjištění, jak „těsně“ zvolená regresní funkce předpokládanou funkční závislost mezi závisle a nezávisle proměnnou vystihuje.“ ([5], str. 102)

Index determinace

Jak již bylo řečeno v předcházejících odstavcích, je nutné vypočítat tzv. *index determinace*, který nabývá hodnot na intervalu $< 0, 1 >$, přičemž čím blíže se hodnota indexu determinace blíží k 1, tím lépe daná regresní funkce onu závislost mezi závisle a nezávisle proměnnou vystihuje. Tento index je vyjádřen předpisem

$$I^2 = \frac{S_{\hat{\eta}}}{S_y}, \quad (1.31)$$

kde $S_{\hat{\eta}}$ je tzv. „rozptyl vyrovnaných hodnot“ (vz. 1.32) a S_y „rozptyl empirických hodnot“ (vz. 1.33).

$$S_{\hat{\eta}} = \frac{\sum_{i=2}^n (\hat{\eta}_i - y_i)^2}{n}, \quad (1.32)$$

$$S_y = \sum_{i=1}^n (y_i - \bar{y})^2. \quad (1.33)$$

1.2.6 Predikce

Při popisu trendu časových řad pomocí regresních funkcí nás nezajímá pouze dosavadní vývoj (trend), ale v mnoha případech potřebujeme vědět, jak se bude trend dále vyvíjet, a proto můžeme pomocí regresních funkcí i hodnoty časové řady tzv. *předpovídat*, resp. *predikovat*. Pro relevantní predikci musíme samozřejmě zvolit i vhodnou regresní funkci pro vyrovnání dat (viz. předchozí podkapitola), protože ne všechny funkce a metody jsou pro predikci dat vhodné.

Typickým případem nepoužitelnosti pro výpočet prognózy je např. výše zmíněná *metoda klouzavých průměrů*, neboť je založena na principu vyrovnávání sousedních hodnot v daném *okénku*, tudíž je možné odhadnout pouze hodnotu následující za posledním prvkem časové řady, což samozřejmě nemá dostatečný informační charakter a nemůžeme hovořit o prognóze.

Proto budu při tvorbě aplikace, konkrétně v části týkající se predikce hodnot časové řady, implementovat pouze tyto regresní funkce :

- Logistický trend.
- Gompertzova křivka.
- Modifikovaný exponenciální trend.
- Regresní přímka.

1.3 Softwarová aplikace

1.3.1 Základní předpoklady

Jak již bylo řečeno v úvodu, cílem této práce je navrhnout a vytvořit aplikaci zaměřenou na zpracování a vizualizaci nejrozumnějších ekonomických dat, zejména však

časových řad. Dílčím cílem této aplikace je být dostatečnou pomůckou při zpracování tohoto typu dat, tedy být uživatelsky nenáročnou a přehlednou k dosažení požadovaných výsledků pro efektivní analýzu.

Z tohoto předpokladu vychází i volba vhodného technologického zpracování, včetně výběru programovacího jazyka, jeho veškerých podpůrných nástrojů a také programového vybavení k jeho implementaci.

Tato aplikace bude tvořena a laděna výhradně pro prostředí Microsoft Windows v kterékoli verzi operačních systémů počínaje Windows XP s opravným balíčkem SP2.

1.3.2 Programovací jazyk

V tomto ohledu mám na výběr mezi dvěma programovacími jazyky, kterými jsou *Sun Java* a *Microsoft C#*. Volba mezi nimi záleží jenom na vývojáři, neboť oba jsou prakticky stejně schopné, až na drobné technologické odchylky, které jsou pro tuto práci nepodstatné a proto zde uvedu pouze základní rozdíly.

Java

Jedná se o poměrně starý a hojně používaný objektově orientovaný jazyk, který si za dobu své existence našel své uplatnění a v dnešní době se s ním setkáváme zejména v náročných internetových aplikacích a rozsáhlých internetových portálech. Jeho největší předností je snadná a většinou bezproblémová přenositelnost na různé platformy a dostupné programové vybavení pro vývoj a ladění všech aplikací.

Pro vývoj a spouštění programů na jakékoliv platformě je však potřeba do operačního systému doinstalovat volně dostupnou komponentu *Java Runtime Environment*, příp. pro vývoj i *Java Development Kit*.

C#

Programovací jazyk *C#* z rodiny Microsoft .NET je jakousi odpovědí na jazyk *Java*, což je jasné vidět při pohledu na zdrojové kódy, kde je celkem zřejmá podobnost, avšak v této práci nebudu rozebírat postup Microsoftu při vývoji tohoto jazyka. Jeho nespornou výhodou oproti Javě je otevřenost ze strany Microsoftu k vývojářům, což obsahuje rozsáhlou dokumentaci dostupnou on-line a také „naslouchání“ jejich potřebám, což přináší v každé nové verzi mnohá užitečná vylepšení. Nevýhodou může být problémová, v podstatě mizivá možnost přenositelnosti na jiné platformy než MS Windows.

Pro vývoj a spouštění aplikací musíme do operačního systému doinstalovat tzv. „.NET Framework“, který je také volně dostupný a dokonce v posledních verzích Microsoft Windows přímo integrován již při instalaci, což jeho použití značně ulehčuje.

Výběr

Společným faktorem obou jazyků je, že pro vývoj, spouštění a běh aplikací musíme do operačního systému doinstalovat jejich potřebné komponenty. Ovšem v případě jazyka *C#* je situace ulehčena integrací jeho potřebných komponent přímo do operačního systému. Jak jsem v úvodu této kapitoly uvedl, bude aplikace určena pro platformu MS Windows, proto tedy pro vývoj aplikace zvolím *C#* s jeho .NET Frameworkem ve verzi 3.5, která byla v době psaní této práce aktuální.

1.3.3 Nástroje

Pro vývoj aplikací se běžně používá tzv. „IDE“ (angl. Integrated Development Environment), které spojuje jednak úpravu a překlad zdrojového kódu včetně zvýraznění syntaxe jazyka a také, v případě tvorby grafických klientských aplikací, i grafický editor pro návrh jejího vzhledu.

Pro zvolený jazyk *C#* v dnešní době existuje mnoho vývojových nástrojů, jejichž použití se liší podle možností a požadavků vývojářů. Nejpoužívanější nástroj pochází přímo z dílny Microsoftu, a to *Microsoft Visual Studio*, jež se na trhu distribuuje v několika verzích podle jeho zaměření a obsahu. Pro účely této práce plně postačí verze určená pro studijní a seznamovací účely s tímto jazykem, která se nazývá *Microsoft Visual C# 2008 Express Edition* a je k dispozici volně ke stažení z internetových stránek společnosti Microsoft.

1.3.4 Použité technologie

V této kapitole uvedu přehled a popis základních technologií použitých při tvorbě aplikace.

Grafický subsystém

Framework .NET poskytuje 2 systémy pro grafická rozhraní aplikací, a to *Windows Forms* a *Windows Presentation Foundations*. První zmiňovaný je starší systém užívaný od počátku .NET frameworku a druhý, tzv. „WPF“, je k dispozici od verze 3.5 a je užíván jako implicitní zobrazovací systém na operačních systémech Windows Vista a Windows 7. Z uvedeného vyplývá, že kvůli kompatibilitě bude pro tuto práci vhodnější zvolit první možnost, tedy *Windows Forms*.

MDI

Angl. *Multiple Document Interface*. Tato technologie umožňuje spouštět a pracovat s více „dětskými“ okny v jednom mateřském. V tomto případě to znamená, že uživatel bude moci pracovat s více daty najednou a tak např. srovnávat 2 a více časových řad bez nutnosti každou otevírat zvlášť.

Multi-threading

V dnešní době je dle mého názoru převážná většina osobních počítačů vybavena vícejádrovými procesory, které jsou schopné zpracovat nejméně 2 tzv. „vlákna“ najednou a tím výrazně urychlit veškeré výpočty, nebo je u nich tato možnost alespoň

napodobena pomocí různých technologií.

Každé „vlákno“ (angl. *thread*) tak zpracovává specifické části kódu programu, které mají být provedeny během jednoho výpočetního cyklu. V případě jednojádrových procesorů bez optimalizací to znamená, že veškeré operace jsou prováděny v jednom vlákně, tzn. za sebou, což se pro uživatele může projevit tzv. „zamrznutím obrazovky“. Ve skutečnosti však program pracuje, ale kvůli náročnosti výpočtu vynaloží veškeré výpočetní prostředky na danou operaci a na provoz aplikace tak prakticky žádné nezbydou, a proto se pak jeví jako nečinná. S nástupem nových technologií optimalizací, či dokonce vícejádrových procesorů, se tak v posledních letech otevírá velice užitečná možnost zkrácení výpočetního času pomocí tzv. „multithreadingu“, tedy možnosti zpracovávat více vláken paralelně a výrazně tím urychlit běh aplikace.

V aplikaci pro tuto bakalářskou práci budu implementovat 2 samostatná vlákna, která budou procesorem zpracovávána souběžně. První vlákno bude vyhrazeno pouze pro zobrazování celého grafického uživatelského rozhraní, tzv. „GUI“ (angl. *graphical user interface*) a druhé, řekněme „pracovní“ vlákno, bude sloužit pro všechny ostatní, dlouhotrvající operace. Jedná se například o veškeré matematické výpočty, analýzy časových řad nebo vykreslování grafů.

Pro správu pracovního vlákna využijeme pomocnou třídu *BackgroundWorker* ze jmenného prostoru *System.ComponentModel*, která dokonale poskytuje základní řízení vlákna (spouštění, ukončování, zachycování výjimek, informace o průběhu zpracování, apod.).

Vykreslování grafů

Pro vykreslování všech grafů je použita komponenta *Microsoft Charts for .NET Framework*, která je volně dostupná ke stažení na webových stránkách společnosti Microsoft. Tato komponenta podle mého názoru poskytuje dostatečné možnosti pro práci s grafy.

Účelem aplikace je efektivně zobrazit zpracovávanou časovou řadu, k čemuž by měly podle mého názoru postačit 2 grafy. Jeden pro komplexní zobrazení původních a vyrovnaných hodnot a druhý pro zobrazení vypočteného trendu spolu s prognózovanou hodnotou.

XML

K ukládání uživatelsky vytvořených časových řad bude sloužit propracovaný značkovací systém XML (angl. „Extensible Markup Language“, jehož obrovskou výhodou je univerzálnost použití a rozsáhlá podpora ze strany snad všech programovacích jazyků a nástrojů.

Ve zvoleném programovacím jazyce je obsaženo mnoho tříd pro práci s XML, tudíž by neměl být problém s jejich implementací pro účely tvořené aplikace.

Tisk

Pro tisk výstupní sestavy bude vytvořena vlastní třída, která bude využívat třídy zabalené ve jmenném prostoru *System.Drawing.Printing*. Uživatel si tak bude moci v přehledném dialogu nastavení vybrat vlastnosti a charakteristiky časové řady, které si přeje vytisknout.

2 IMPLEMENTACE APLIKACE

Tato kapitola byla zpracována za pomoci literárních zdrojů [2] a [6].

2.1 Vrstvy aplikace

2.1.1 Aplikační vrstva

Do aplikační vrstvy programu zařadím veškeré třídy pro práci s časovými řadami, tj. třídy pro vytváření, otevírání, ukládání, zpracování, predikci a vizualizaci časových řad.

2.1.2 Datová vrstva

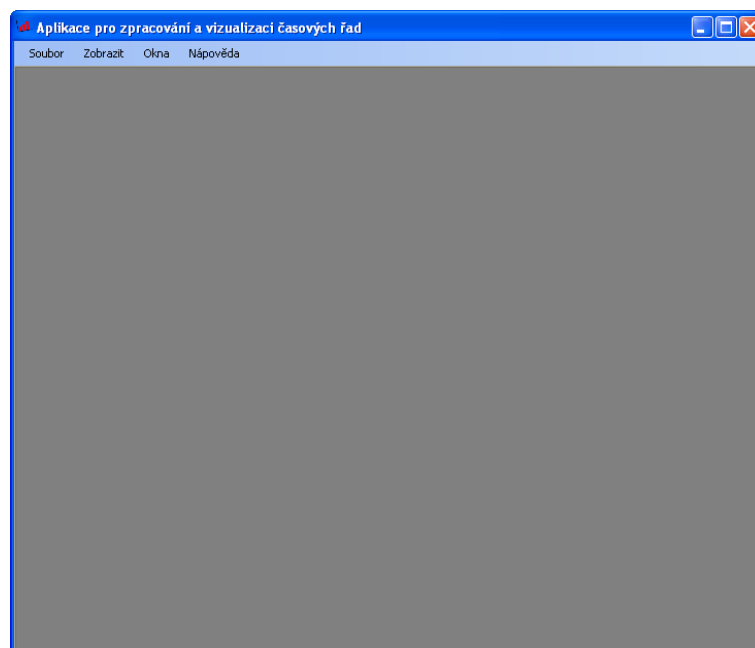
Datová vrstva vytvářené aplikace je založena na operacích se soubory XML (viz. předchozí kapitola). Pro práci se soubory XML budu používat třídy a jejich metody obsažené ve jmenném prostoru *System.Xml*. Pro uložení hodnot a vlastností časové řady jsem navrhl tuto strukturu dokumentu:

```
<?xml version="1.0" standalone="yes"?>
<Název časové řady>
<TS>
<x>hodnota</x>
<y>hodnota</y>
</TS>
...
..
<Vlastnosti>
<typ>Typ</typ>
<axisX>Název nezávisle proměnné</axisX>
<axisY>Název závisle proměnné</axisY>
</Vlastnosti>
</Název časové řady>
```

Při ukládání do XML souboru dojde k serializaci hodnot z polí pro zadávání dat a uložení do bloku *Název časové řady* po dvojicích $\langle x \rangle \text{ hodnota} \langle /x \rangle$ a $\langle y \rangle \text{ hodnota} \langle /y \rangle$.

2.1.3 Prezentační vrstva

Prezentační vrstva aplikace představuje tzv. „grafické uživatelské rozhraní“, které musí být navrženo tak, aby bylo pro uživatele jednoduše a přehledně obsluhovatelné s důrazem na efektivní rozložení ovládacích prvků. Jedná se především o to, aby se uživatel při práci s programem nemusel zdržovat zmateným „blouděním“ po aplikaci a neztrácel tak čas.



Obrázek 2.1: Hlavní okno aplikace, Zdroj: vlastní

2.2 Jmenné prostory a třídy

Veškeré vlastní třídy vytvořené pro realizaci aplikace jsou zabalené do jmenného prostoru *Thesis*, přičemž tento je pro přehlednost rozdělen do dílčích jmenných prostorů podle jejich významu. V následujícím textu jsou uvedeny jmenné prostory s příslušnými třídami.

2.2.1 Thesis

V tomto jmenném prostoru je umístěna pouze jedna třída *Program* obsahující statickou metodu *Main()*, která je volána vždy při startu aplikace. Nazývá se tzv. „vstupním bodem programu“.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using Thesis.Forms;
namespace Thesis
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FormMain());
        }
    }
}
```

2.2.2 Thesis.Core

Tento jmenný prostor zabaluje třídy pro provádění veškerých výpočetních operací, které nejsou pro uživatele tzv. „viditelné“, tj. ty, které běží v dříve popsaném pracovním vláknu.

Prediction

Metody třídy *Prediction* jsou volány bez vytváření instance této třídy. Obsahují tedy statické metody, např. *ChooseBalanceFunction(TimeSeries timeseries)*, které jsou užité pro automatický výběr vyrovnávací funkce. Obdobnou funkci má statická metoda *ChoosePredictFunction()*, která je určena pro automatický výběr funkce pro prognózu hodnot.

```
public static int ChooseBalancingFunction(TimeSeries timeseries)
{
    ....
}
```

Statické metody jsou uvozeny klíčovým slovem **static** a nemohou mít stejný název jako název třídy. Ve shora uvedeném zdrojovém kódu je nastíněna výše uvedená metoda pro volbu vyrovnávací funkce. Předává se jí vstupní parametr typu **TimeSeries**, což je vytvořená časová řada (viz. níže) a vrací číselný kód číselného typu **int** vyrovnávací funkce.

PrintPage

Třída je tzv. „statická“, to znamená, že můžeme používat její metody i bez vytvoření její instance. Díky tomu můžeme pohodlně tisknout a nevytvářet zbytečně další instanci. Jelikož si uživatel může sám zvolit, které vlastnosti chce tisknout, musí třída obsahovat metodu pro výpočet velikosti tabulky a rovněž rozměrů stránky, případně i počet stránek potřebných k vytištění celé sestavy. Ukázka tiskového výstupu v příloze č. 3 - Tiskový výstup.

Settings

Tato třída je odvozena od базové třídy *ApplicationSettingsBase* a obsahuje jak metody svojí базové třídy, tak nově vytvořené metody pro práci s uživatelsky nastavenými vlastnostmi aplikace, jako jsou např. název tiskárny, zvolené vlastnosti a charakteristiky časových řad. Následující ukázka zdrojového kódu demonstruje deklaraci přístupových vlastností členské proměnné *PrinterName*, ve které je uložen právě název tiskárny.

Vlastnost **get** zajišťuje „získání“ hodnoty z proměnné *PrinterName*, kdežto vlastnost **set** obsluhuje uložení hodnoty do této proměnné.

```

[UserScopedSetting()]
[DefaultSettingValue("")]
public string PrinterName
{
    get
    {
        return ( ( string )this["PrinterName"] );
    }
    set
    {
        this["PrinterName"] = ( string )value;
    }
}

```

TimeSeries

Tato třída obstarává vytvoření časové řady z XML souboru. Při zavolání této třídy dojde k vytvoření generických kolekcí s hodnotami nezávisle proměnné textového typu *string* a hodnotami závisle proměnné číselného *double*, tedy desetinných čísel s dvojitou přesností. Třída obsahuje i metody pro výpočet všech charakteristik a jejich zápis do příslušných polí.

Pro příklad uvedu implementaci vzorce pro výpočet neváženého chronologického průměru (1.25) v privátní metodě *CountSampleMean()*.

```

double midSum = 0;
double first = _originalDataY[0] / 2; ;
double last = _originalDataY[( _itemsNo - 1)] / 2;

for (int i = 1; i < ( _itemsNo - 1); i++)
{
    midsuma += _originalDataY[i];
}

this._sampleMeanY = Math.Round((first + midSum + last) / ( _itemsNo - 1), 4);

```

2.2.3 Thesis.Core.Functions

Tento jmenný prostor zabaluje třídy pro matematické a statistické funkce. Obsahuje následující třídy :

Exponencial

Z třídy *Exponencial* budou odvozeny další třídy sloužící pro vyrovnání dat pomocí exponenciálních funkcí (Logistický trend, Gompertzova křivka, Modifikovaný exponenciální trend). Obsahuje tak metody pro výpočet odhadu koeficientů regresních funkcí, neboť tyto funkce se ve výpočtu liší pouze u sum. Tato třída tedy obsahuje členské proměnné typu generické kolekce desetinných čísel s dvojitou přesností.

```
private List<double> _originalDataY = new List<double>();
public Exponencial(List<double> data)
{
    _originalDataY = data;
}
```

V ukázce zdrojového kódu je znázorněna deklarace privátní členské proměnné *_originalDataY* a konstruktor třídy se vstupním parametrem kolekce s daty časové řady, který se zavolá při vytváření instance a uloží data do svojí členské proměnné.

Jelikož tyto exponenciální funkce musí mít pro určení odhadu regresní funkce počet hodnot dělitelný 3, musel jsem vytvořit metodu pro vytvoření nových hodnot. Při zjištění, že zbytek po dělení počtu původních hodnot třemi je nenulový, odebere příslušný zbytek od konce časové řady.

```
protected List<double> NewDataList()
{
    List<double> output = new List<double>(_originalDataY);
    int origPocet = output.Count;
    int zbytek = ( origPocet % 3 );
    if ( zbytek != 0 )
    {
        output.RemoveRange(origPocet - zbytek, zbytek);
    }
    return output;
}
```

LogisticTrend

První třída odvozená od třídy *Exponencial*. Při vytváření instance se předá konstruktoru vstupní parametr (generická kolekce s hodnotami časové řady) a zároveň odkaz na příslušný konstruktor báze třídy.

```
private List<double> _workingData = new List<double>();

public LogisticTrend(List<double> data)
: base(data)
{
    _workingData = base.NewDataList();
    base.groupNo = ( this._workingData.Count / 3 );
    this.CountSums();
    base.CountApproximationIndices();
}
```

Gompertz

Třída pro vyrovnaní dat pomocí Gompertzovy křivky. Struktura metod je stejná jako předchozí, liší se pouze v matematickém výpočtu sum.

ModifiedExponencialTrend

Třída pro vyrovnaní dat pomocí Modifikovaného exponenciálního trendu. Popis třídy je shodný s předchozí.

MovingAverages

Tato třída slouží pro vyrovnaní hodnot časové řady pomocí klouzavých průměrů. Obsahuje metody pro výpočet vyrovnané hodnoty (postup viz. výše). Příkladem metody pro výpočet sloupcového vektoru hodnot \mathbf{c}_k dle vz. (1.19) je následující výpis kódu.

```

public double[,] ValuesColumnVector(int position)
{
    int i, n = 0;
    double C11 = 0;
    double C12 = 0;
    double C13 = 0;
    double C14 = 0;

    if ( position == 0 )
    {
        i = 0;
        n = i + 4;
    } else {
        n = (this._dataY.Count-1);
        i = n - 4;
    }
    for ( int a = i ; a < n+1 ; a++ )
    {
        C11 += this._dataY[a];
    }

    C12 = ( ( ( -2 ) * this._dataY[i] ) + ( ( -1 ) * this._dataY[i+1] ) + (this._dataY[i+3])
    + ( 2 * this._dataY[i+4] ) );

    C13 = ( ( Math.Pow(-2, 2) * this._dataY[i] ) + ( Math.Pow(-1, 2) * this._dataY[i+1] )
    + this._dataY[i+3] + ( Math.Pow(2, 2) * this._dataY[i+4] ) );
    C14 = ( ( Math.Pow(-2, 3) * this._dataY[i] ) + ( Math.Pow(-1, 3) * this._dataY[i+1] )
    + this._dataY[i+3] + ( Math.Pow(2, 3) * this._dataY[i+4] ) );

    double[,] CK = {{C11}, {C12}, {C13}, {C14} };
    return CK;
}

```

Z typografických důvodů jsou některé výpočty rozděleny do více řádků.

Regression

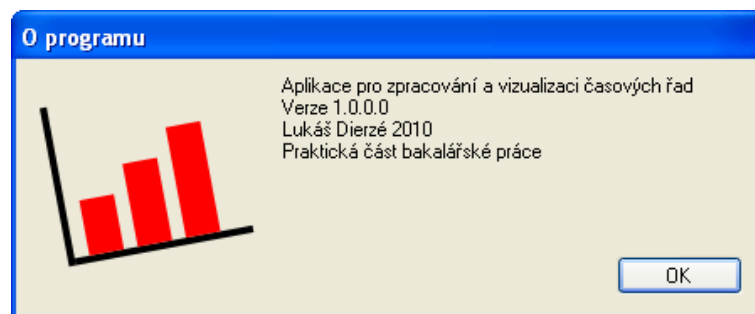
Třída *Regression* je implementací metody vyrovnání dat pomocí *Regresní přímky*. Obsahuje metody pro výpočet výběrových průměrů dle vz. (1.8) a výpočet odhadů koeficientů dle vz. (1.7).

2.2.4 Thesis.GUI

Zde jsou zabaleny třídy, které obstarávají běh grafického uživatelského rozhraní.

FormAboutBox

Třída, která při zavolání ukáže okno s automaticky aktualizovanými informacemi o programu. Tato třída má pouze informativní charakter a neobsahuje žádné významné metody.



Obrázek 2.2: Třída FormAboutBox - O programu, Zdroj: vlastní

FormSettings

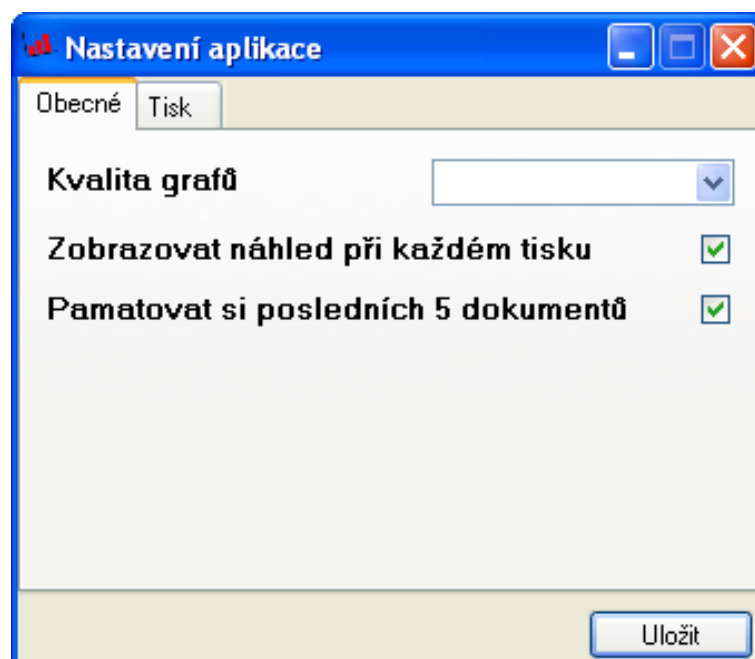
Třída definující okno pro práci s nastavením. Pomocí této třídy může uživatel interaktivně nastavovat některé vlastnosti třídy. Při zavolání si třída načte uložené vlastnosti a jejich hodnoty zobrazí. Při změně vlastností musí uživatel tyto uložit, přičemž po uložení třída restartuje aplikaci, aby došlo ke správnému načtení. O tomto procesu je uživatel informován dialogovým oknem.

FormTimeSeries

Tato třída má na starosti kompletní zobrazení časové řady a jejích vlastností. Při zavolání této třídy dojde k vytvoření instance třídy *TimeSeries*, která vytvoří časovou řadu a zapíše ji do připraveného pole.

Třída obsahuje metody pro překreslování velikostí prvků při změně velikosti okna, neboť díky technologii MDI může uživatel inicializovat několik těchto tříd najednou a poté je potřeba okna uspořádat, tedy změnit jejich velikosti apod.

Z této třídy dochází k volání třídy *Prediction*, resp. jedné z tříd pro vyrování dat (výše uvedené). K zavolání těchto tříd dojde jakmile uživatel zvolí některou z možností pro vyrovnaní dat.



Obrázek 2.3: Třída FormSettings - Nastavení aplikace, Zdroj: vlastní

FormTsWizard

Třída definující okno pro vytváření nové časové řady. Obsahuje textová pole, přepínače a DataGridView pro zadávání dat a metody pro jejich obsluhu. Při zadávání dat třída kontroluje, zda uživatel správně zadal data a v případě chyby, nebo nedostatků jej o tom varuje pomocí textového upozornění.

Testovací data

Časová řada Charakteristiky a Prognózy Grafy

Název : **Testovací data**
 Typ : **Intervalová časová řada**
 Regresní funkce :

i	roky	počet studentů v prezenční formě	První difference	Koef. růstu	Vyrovnané hodnoty
1	1992	16069	N/A	N/A	
2	1993	14908	-1161	0,93	
3	1994	16639	1731	1,12	
4	1995	16682	43	1	
5	1996	18398	1716	1,1	
6	1997	20942	2544	1,14	
7	1998	23262	2320	1,11	
8	1999	23582	320	1,01	
9	2000	24024	442	1,02	
10	2001	24866	842	1,04	
11	2002	25865	999	1,04	
12	2003	26440	575	1,02	
13	2004	29950	3510	1,13	

Obrázek 2.4: Třída FormTimeSeries, Zdroj: vlastní


Průvodce vytvořením nové časové řady

Základní charakteristiky

Název časové řady :
 Nezávisle proměnná :
 Závisle proměnná :

☐ Intervalová
☐ Okamžiková

Data



Zadejte název časové řady

Obrázek 2.5: Třída FormTsWizard - Průvodce vytvořením časové řady, Zdroj: vlastní

3 TESTOVÁNÍ APLIKACE

3.1 Testovací data

Pro otestování vytvořené softwarové aplikace vhodně poslouží data ze statistické ročenky Českého statistického úřadu, týkající se průměrných hrubých mezd v letech 2000 až 2009 v průměru za celou Českou republiku, tzn. nebudeme rozlišovat průměrné mzdy podle odvětví, podnikatelské/nepodnikatelské sféry apod.

Rok	Nominální mzda [Kč]
2000	13 219
2001	14 378
2002	15 524
2003	16 430
2004	17 466
2005	18 344
2006	19 546
2007	20 957
2008	22 691
2009	23 598

Tabulka 3.1: Průměrná hrubá měsíční mzda v ČR, Zdroj: [8]

3.2 Vytvoření časové řady

Pro vytvoření časové řady je v tomto případě třeba zvolit ruční zadání dat (viz. příloha č. 1 : Uživatelská příručka, sekce 3a. Vytváření časové řady). Zadáním hodnot (název, nezávisle a závisle proměnná, typ) se zpřístupní pole pro zadávání dat. Po zadání hodnot z tabulky 3.1 klikneme na tlačítko **Uložit**, vybereme cestu k souboru a po úspěšném uložení můžeme s časovou řadou ihned pracovat.

Průvodce vytvořením nové časové řady

Základní charakteristiky

Název časové řady:

Nezávisle proměnná:

Závislé proměnná:

☒ Intervalová ☐ Okamžiková

Data

	Nezávisle proměnná	Závisle proměnná
	2001	14378
	2002	15524
	2003	16430
	2004	17466
	2005	18344
	2006	19546
	2007	20957
	2008	22691
	2009	23598
	*	

Obrázek 3.1: Ukázka vytvoření testovací časové řady, Zdroj: vlastní

3.3 Zpracování a vizualizace dat

3.3.1 Výpočet charakteristik a vyrovnání dat

Ihned po otevření časové řady se provede výpočet těchto charakteristik, přičemž jejich hodnoty lze nalézt na záložkách **Časová řada** a **Charakteristiky**.

Charakteristika	Hodnota
Počet prvků	10
Minimální hodnota	13 219
Maximální hodnota	23 598
Výběrový průměr	18 215,3
Průměr prvních diferencí	1 153,2
Průměr koeficientů růstu	1,0597

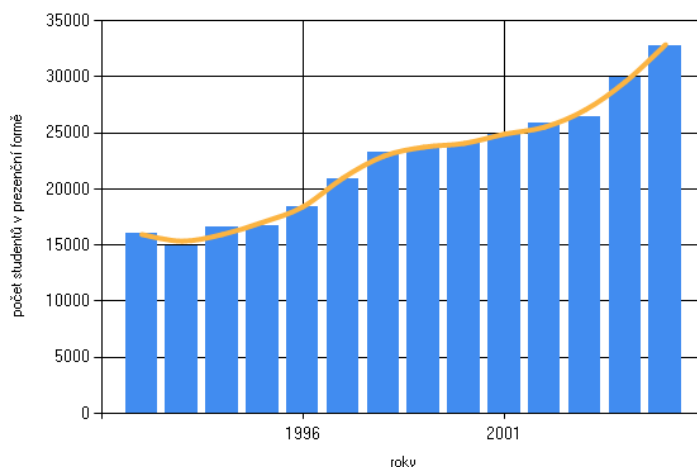
Tabulka 3.2: Vypočtené charakteristiky testovací časové řady, Zdroj: vlastní

Nyní nás zajímá trend této časové řady, a proto v roletce **Regresní funkce** vybereme *Automaticky zvolit nejvhodnější funkci*. Program v tomto případě automaticky zvolí *Metodu klouzavých průměrů* a vyrované hodnoty zapíše do tabulky v

záložce **Časová řada** do sloupce *Vyrovnané hodnoty* a spočtený index determinace zobrazí v záložce **Charakteristiky**, jež se v tomto případě rovná hodnotě 0,9997.

3.3.2 Vykreslení grafu

Graf s původními a vyrovnanými hodnotami časové řady (viz. Příloha 1 : Uživatelská příručka, sekce 3e. Grafy) nalezneme v záložce **Graf**.



Obrázek 3.2: Graf testovací časové řady, Zdroj: vlastní

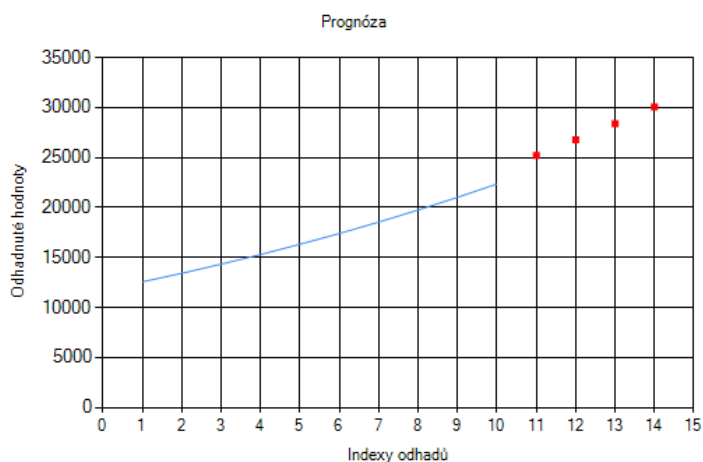
Na tomto obrázku je *komplexní graf* analyzované časové řady včetně vyrovnaných hodnot, v tomto případě pomocí automaticky zvolené *Metody klouzavých průměrů* (viz. výše).

V dalším kroku můžeme prognózovat budoucí vývoj. Z bližšího zkoumání trendu časové řady můžeme usoudit, že pro prognózu můžeme použít buď *Logistický trend*, nebo *Regresní přímku*.

3.4 Prognóza

V záložce *Prognóza* proto vyplníme regresní funkci a rozsah hodnot (viz. Příloha A : Uživatelská příručka, sekce 3f. Prognózy). Volbu regresní funkce necháme na programu a řekneme, že nás zajímá odhad průměrných mezd v letech 2010 až 2013,

tj. do rozsahu hodnot zadáme číslo 4 a klikneme na **Prognóza**.



Obrázek 3.3: Graf prognózy, Zdroj: vlastní

Program nejdříve vybere vhodnou regresní funkci (v tomto případě *Logistický trend*) a poté vypočte prognózy v požadovaném rozsahu. Jejich hodnoty ihned zobrazí na grafu pod formulářem (obr. 3.3). Najetím myši na zakreslené body se ukáže předpovězená hodnota.

Index	Rok	Hodnota
11	2010	25 220,54
12	2011	26 757,89
13	2012	28 364,70
14	2013	30 041,20

Tabulka 3.3: Prognóza průměrných mezd v letech 2010 až 2013, Zdroj: vlastní

Z uvedené tabulky prognóz pro daný rozsah vyplývá, že odhadnutá hodnota průměrné mzdy roce 2010 je 25 220,54 Kč.

3.5 Časová náročnost výpočtu

Časová náročnost kompletního výpočtu charakteristik časové řady, jejího vyrovnaní a vykreslení všech grafů je měřená ve vteřinách od spuštění analýzy po zobrazení kompletního výstupu, přičemž se nezapočítávají interakce s programem během procesu výpočtu (kontrolní dialogy). Z uvedeného vyplývá, že časová náročnost výpočtu při analýze časových řad bude jistě záviset především na rozsahu časové řady a jejích hodnotách, ale samozřejmě i na výpočetním výkonu počítače.

Operace	Čas
Vytvoření časové řady	< 0,1 s
Zpracování dat	< 0,1 s
Výpočet charakteristik	< 0,1 s
Vyrovnaní dat	< 0,1 s
Vykreslení grafů	< 0,1 s
Celkem	$\cong 0,5$ s

Tabulka 3.4: Časová náročnost analýzy časové řady, Zdroj: vlastní

V tabulce je uvedena časová náročnost analýzy testovací časové řady definované v této kapitole, přičemž všechny hodnoty jsou běžným uživatelem prakticky neměřitelné, proto je u všech položek uvedena přibližná hodnota.

4 NÁVRHY NA ROZŠÍŘENÍ

V této kapitole jsou uvedeny některé možnosti budoucího rozšíření aplikace.

Statistické funkce

Aplikaci by dále bylo možné rozšířit o další funkce týkající se především složitějších statistických metod, např. analýzy zahrnující sezónní vlivy, další regresní funkce apod.

Nápověda

S dalším rozšiřováním aplikace souvisí i větší rozsah možností, tudíž by bylo nutné poskytnout uživateli podrobnější podporu ve formě nápovědy buď integrované do programu, nebo dostupnou on-line na internetu, či ke stažení, např. ve formátu PDF.

Konzole

Do aplikace je možné i doplnit třídu pro zobrazení tzv. „konzole“, která by mohla významně urychlit proces analýzy dat. V současné podobě aplikace je totiž nutné zadávat časové řady pomocí grafického rozhraní, což zajistí uživatele do jisté míry zdržuje. Navrhovaná konzole by tak umožnila zadávat data a jejich parametry přímo pomocí příkazů tak, jak to umožňuje například z programů *Maple* nebo *Matlab*.

Aktualizační systém

S možnostmi různých rozšíření je třeba udržovat aplikaci stále aktualizovanou, ať už kvůli novým přídavkům nebo záplatám možných chyb v programu. Jednalo by se tak o třídu pro kontrolu současné verze a případné aktualizace novějších souborů. Proces aktualizace by probíhal buď automatickým stažením aktualizovaných dat z internetového úložiště vyhrazeného pro tuto aplikaci, nebo ručním stažením a instalací opravného balíčku.

ZÁVĚR

Cílem této práce bylo vytvořit aplikaci pro zpracování a vizualizaci časových řad. Při tvorbě této aplikace jsem nejprve popsal teoretické předpoklady pro práci s časovými řadami a poté jsem musel rozhodnout, jaké dostupné prostředky a technologie využiji pro samotnou realizaci. Vzhledem k osobním preferencím a zkušenostem jsem zvolil programovací jazyk *c#*.

V souladu s jasně daným cílem jsem navrhl celkovou strukturu programu, podle které jsem vytvářel všechny potřebné třídy a metody tak, aby jejich použití bylo co nejefektivnější. Cílem mé usilovné práce je výsledná aplikace, jejíž použití je dle mého názoru dostatečně intuitivní a veškeré výstupy mají potřebný, předem definovaný výstupní charakter, tj. zpřístupnit statistické metody při analýze dat podnikovým manažerům a tím jim poskytnout silnou oporu pro jejich práci.

V předcházející kapitole jsem shrnul možná budoucí rozšíření programu, neboť jejich realizace nebyla při tvorbě přímo potřebná a neodpovídala mým základním požadavkům na funkčnost aplikace.

LITERATURA

- [1] ANDĚL, J. *Statistické metody*. 2. vyd. Praha : MATFYZPRESS. 1998. ISBN 80-85863-27-8.
- [2] BAYER, J. *C# 2005 Velká kniha řešení*. 1. vyd. Brno : Computer Press, 2007. ISBN 978-80-251-1620-3.
- [3] CIPRA, T. *Analýza časových řad s aplikacemi v ekonomii*. 1. vyd. Praha : SNTL/ALFA, 1986. ISBN 99-00-00157-X.
- [4] HINDLS, R. a spol. *Statistika pro ekonomy*. 8. vyd. Praha : Professional Publishing, 2007. ISBN 978-80-86946-43-6.
- [5] KROPÁČ, J. *Statistika B*. 2. vyd. Brno : Vysoké učení technické v Brně, 2009. ISBN 978-80-214-3295-6.
- [6] MAREŠ, A. *1001 tipů a triků pro C#*. 1. vyd. Brno : Computer Press, 2008. ISBN 978-80-251-2125-2.
- [7] ZVÁRA, K. *Regresní analýza*. Academia. Praha. 1989. ISBN 80-200-0125-5.
- [8] Průměrná měsíční mzda - ČSÚ [online]. [cit. 2010-05-24]. Dostupné z:
[http://www.czso.cz/csu/csu.nsf/i/tab_1_pmz/\\$File/cpmzcr030910_1.xls](http://www.czso.cz/csu/csu.nsf/i/tab_1_pmz/$File/cpmzcr030910_1.xls)

SEZNAM OBRÁZKŮ

1.1	Znázornění naměřených n dvojic, Zdroj: [5], str. 79	12
1.2	Metoda nejmenších čtverců, Zdroj: [5], str. 80	14
1.3	Okénko pro klouzavé průměry, Zdroj: [5], str. 126	18
1.4	Sloupcový, hůlkový a spojnicový graf, Zdroj: vlastní	20
2.1	Hlavní okno aplikace, Zdroj: vlastní	32
2.2	Třída FormAboutBox - O programu, Zdroj: vlastní	39
2.3	Třída FormSettings - Nastavení aplikace, Zdroj: vlastní	40
2.4	Třída FormTimeSeries, Zdroj: vlastní	41
2.5	Třída FormTsWizard - Průvodce vytvořením časové řady, Zdroj: vlastní	41
3.1	Ukázka vytvoření testovací časové řady, Zdroj: vlastní	43
3.2	Graf testovací časové řady, Zdroj: vlastní	44
3.3	Graf prognózy, Zdroj: vlastní	45

SEZNAM TABULEK

3.1	Průměrná hrubá měsíční mzda v ČR, Zdroj: [8]	42
3.2	Vypočtené charakteristiky testovací časové řady, Zdroj: vlastní	43
3.3	Prognóza průměrných mezd v letech 2010 až 2013, Zdroj: vlastní . . .	45
3.4	Časová náročnost analýzy časové řady, Zdroj: vlastní	46

SEZNAM POUŽITÝCH ZKRATEK

angl. anglicky

apod. a podobně

atd. a tak dále

ČSÚ Český statistický úřad

GDI Graphics Device Interface

GUI Graphical User Interface

MDI Multiple Document Interface

např. například

obr. obrázek

PDF Portable Document File

příp. případně

resp. respektive

tj. to jest

tzn. to znamená

XML Extensible Markup Language

SEZNAM PŘÍLOH

Příloha č. 1 : Uživatelská příručka

Příloha č. 2 : Obsah přiloženého DVD

Příloha č. 3 : Tiskový výstup

PŘÍLOHA Č. 1 : UŽIVATELSKÁ PŘÍRUČKA

1. Instalace a spuštění aplikace

1a. Potřebný software

Aplikace je určena pro operační systémy Microsoft Windows XP SP2 a vyšší. Pro běh je třeba mít nainstalován Microsoft .NET Framework ve verzi alespoň 3.5.

1b. Spuštění

Tuto aplikaci není potřeba instalovat, stačí pouze nakopírovat soubor **Thesis.exe** do jakékoli složky na PC a následně jej spustit. Po spuštění se zobrazí hlavní okno aplikace.

2. Nastavení aplikace

Základní nastavení aplikace, jako je výběr obecných tiskových úloh, barevná pozadí datových polí, či velikost písma se provede kliknutím **Nastavení** v záložce **Aplikace** v horním menu. Po upravení všech vlastností a kliknutí na tlačítko **Uložit** dojde k uložení informací a restartu aplikace. Nastavení se ukládá pro každého uživatele PC zvlášť.

3. Analýza časové řady

3a. Vytvoření časové řady

Časovou řadu lze vytvořit kliknutím na **Nový** v záložce **Soubor** nebo pomocí klávesové zkratky CTRL+N. Zobrazí se *Průvodce vytvořením nové časové řady*, kde uživatel vyplní všechna pole dle instrukcí a kliknutím na **Uložit** se zobrazí dialogové okno pro výběr cesty a názvu. Poté se časová řada uloží do zvoleného umístění. Po uložení může uživatel s časovou řadou začít ihned pracovat.

3b. Otevření časové řady

Časovou řadu lze otevřít kliknutím na **otevřít** v záložce **Souboru** nebo pomocí klávesové zkratky CTRL+O. Otevře se dialogové okno pro výběr souboru s časovou řadou. Pokud si uživatel zvolil v nastavení možnost *Pamatovat si posledních 5 dokumentů*, může otevřít časovou řadu kliknutím na název souboru u **Otevřít poslední** v záložce **soubor**.

3c. Operace s časovou řadou

S hodnotami časové řady lze provádět následující operace :

- editace,
- mazání,
- přidávání,

příčemž při každé z uvedených úprav dojde k přepočítání vlastností a překreslení grafů časové řady.

3d. Uložení časové řady

Časovou řadu může uživatel kliknutím na **Uložit** (nebo pomocí klávesové zkratky CTRL+S) ze záložky **Časová řada** buď uložit pod stejným názvem, tj. přepsat, a nebo může zvolit kliknutím na **Uložit jako** zvolit nový název souboru, pod kterým se časová řada uloží, zavře a načte z nového umístění.

3e. Grafy

V záložce *Komplexní graf* se nachází tzv. *komplexní graf*, který obsahuje znázornění časové řady a po zvolení vyrovnávací funkce také křivku této funkce.

V záložce *Prognózy* je umístěn graf prognózovaných dat, tj. obsahuje křivku vyrovnaných dat a po zvolení funkce pro prognózu a rozsahu prognózovaných dat i křivku těchto odhadů (červeně).

Kliknutím pravým tlačítkem myši nad grafem může uživatel graf buď uložit (do formátu .PNG) nebo překreslit (v případě nekorektního zobrazení dat).

Najetím myši nad křivku nebo graf hodnot se zobrazí aktuální hodnota.

3f. Prognózy

V záložce *Prognóza* je umístěn formulář pro výpočet prognózy pro zadaný rozsah dat. Uživatel nejprve vybere regresní funkci, která bude užita pro výpočet prognózy, přičemž má na výběr z těchto funkcí :

- Automaticky zvolit nejvhodnější funkci.
- Gompertzova křivka.
- Logistický trend.
- Modifikovaný exponenciální trend.
- Regresní přímka.

Poté uživatel vyplní rozsah hodnot, které chce prognózovat, tj. počet hodnot od posledního údaje v časové řadě¹. Po korektním vyplnění aplikace vypočte požadované hodnoty a výsledky zobrazí jak textově tak graficky na grafu pod formulářem.

3g. Tisk výstupu

Následující parametry tiskové sestavy uživatel nastavuje pomocí okna Nastavení.

- Tiskárna
- Sloupce
 - Index
 - Nezávisle proměnná
 - Závisle proměnná

¹Při použití Regresní přímky se vyplní hodnoty nezávisle proměnné oddělené středníkem.

- První difference
- Koeficienty růstu
- Vyrovnané hodnoty
- Hodnoty
 - Typ
 - Index determinace
 - Minimální hodnota
 - Výběrový průměr
 - Maximální hodnota
 - Průměr prvních diferencí
 - Průměr koeficientů růstu
 - Počet prvků

Časovou řadu lze vytisknout kliknutím na **Tisk** v záložce **Časová řada** nebo pomocí klávesové zkratky CTRL+P.

3h. Ukončení práce s časovou řadou

Práci s časovou řadou lze ukončit kliknutím na tlačítko **Zavřít** v záložce **Časová řada**, přičemž bude uživatel dotázán, zda chce opravdu časovou řadu uzavřít.

4. Práce s více časovými řadami

Aplikace umožňuje otevřít více časových řad najednou. Postup otevření viz. **Otevření časové řady**. Poté je možné s okny časových řad pracovat tak, jako by byly samostatnými okny, tzn. zmenšovat, minimalizovat apod. Aplikace umožňuje okna v hlavním okně srovnávat, a to následujícími způsoby :

- **Kaskáda** - okna jsou seřazena kaskádovitě podle záhlaví,
- **Srovnat horizontálně** - okna jsou zobrazena pod sebou,

- **Srovnat vertikálně** - okna jsou zobrazena vedle sebe,
- **Uspořádat** - záhlaví minimalizovaných oken jsou uspořádány abecedně.

Tyto možnosti jsou uživateli zpřístupněny v záložce **Okna** v hlavním okně aplikace po otevření alespoň jedné časové řady.

5. Ukončení aplikace

Při ukončení aplikace a alespoň jedné otevřené časové řadě je uživatel dotázán, zda chce aplikaci opravdu ukončit.

PŘÍLOHA Č. 2 : OBSAH PŘILOŽENÉHO DVD

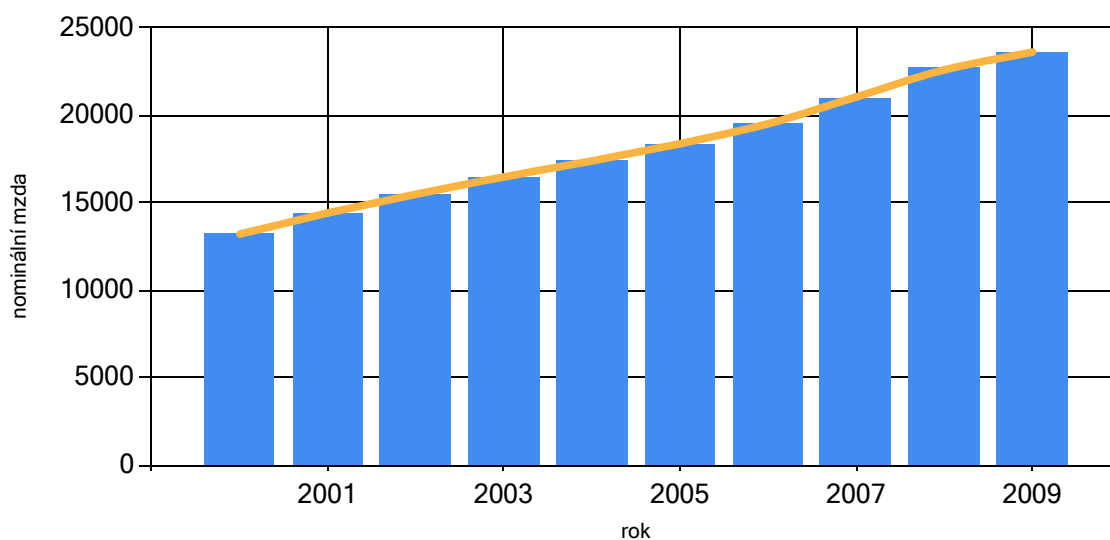
- \aplikace\src\ **Zdrojové soubory aplikace**
- \aplikace\bin\ **Zkompilovaná aplikace**
- \aplikace\frw\ **Instalační soubor .NET Frameworku 3.5**
- \pdf\lukasdierze.pdf **Bakalářská práce ve formátu PDF**

PŘÍLOHA Č. 3 : TISKOVÝ VÝSTUP

Tiskový výstup z aplikace. V nastavení byla zaškrtnuta všechna pole a všechny charakteristiky.

Typ	Intervalová časová řada
Index determinace	0,9997
Minimální hodnota	13219
Výběrový průměr	18215,3
Maximální hodnota	23598
Průměr prvních diferencí	1153,2
Průměr koeficientů růstu	1,0597
Počet prvků	10

i	rok	nominální mzda	První diference	Koef. růstu	Vyrovnané hodnoty
1	2000	13219	N/A	N/A	13210,5
2	2001	14378	1159	1,09	14412,1
3	2002	15524	1146	1,08	15472,8
4	2003	16430	906	1,06	16486,4
5	2004	17466	1036	1,06	17400
6	2005	18344	878	1,05	18395,2
7	2006	19546	1202	1,07	19526,4
8	2007	20957	1411	1,07	21065,3
9	2008	22691	1734	1,08	22618,8
10	2009	23598	N/A	1,04	23616,1



Prognóza

